Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (previously presented) A microcontroller the programming of which is carried out in at least one machine-dependent assembler language in which the assembler commands, with the exception of conditional program jumps or program branches, respectively, are executed independently of data, comprising at least one random number generator assigned to the microcontroller, by means of which the program jumps and program branches are executed in dependence on the state of the random number generator and independently of the internal state of the programming of the microcontroller.

2. (previously presented) A microcontroller as claimed in claim 1, characterized by at least one, in particular bit-addressable, random number register assigned to the random number generator.

3. (previously presented) A microcontroller as claimed in claim 1, characterized by an embodiment as a smartcard controller.

4. (previously presented) An electrical or electronic device controlled by means of at least one microcontroller as claimed in claim 1.

5. (previously presented) A method for processing the programming of a microcontroller executed in at least one machine-dependent assembler language, the assembler commands, with the exception of conditional program jumps or branches, being executed essentially independently of data, characterized in that the program jumps

or program branches are executed in dependence on the state of at least one random number generator and/or independently of the internal state of the programming of the microcontroller.

6. (previously presented) A method as claimed in claim 5, characterized in that the random number generated by the random number generator read via software via registers and the random number read is then evaluated with a conditional program jump or branch.

7. (previously presented) A method as claimed in claim 5, characterized in that, if at least one, in particular bit-addressable, random number register is present, testing per bit of the random number register and a conditional jump or branch is carried out.

8. (previously presented) A method as claimed in claim 5, characterized by the implementation of at least one assembler command ("branch on random bit"), a defined bit of the random number register being supplied, in particular directly, to a condition input for the conditional jump or branch.

9. (previously presented) A method as claimed in claim 5, characterized in that at least one Arithmetic Logic Unit (ALU) flag controlling the conditional jumps or branches is replaced, in particular via the software, by at least one bit of the random number register, so that the conditional jumps or branches corresponding to the bit of the Arithmetic Logic Unit are controlled by the bit of the Random Number Register.

10. (previously presented) A use of a microcontroller as claimed in claim 1 for completely concealing the programming running on the microcontroller, so that at least one program running on the microcontroller is unpredictable and non-reproducible for an external observer.

11. (previously presented) A microcontroller comprising:

a central processing unit;

a memory accessible to the central processing unit, wherein the memory comprises instructions and wherein the central processing unit is configured for:

accessing the instructions, wherein the instructions comprise different instruction sequences for accomplishing a same desired action and where each different instruction sequence produces a same result value for a same input value;

receiving a random number associated with one of the different instruction sequences;

receiving the same input value; and

executing the one of the different instruction sequences associated with the random number using the same input value to produce the same result.

12. (previously presented) The microcontroller of claim 1, wherein the instructions further comprise a jump instruction subject to a condition which if true directs the central processing unit to a program address, wherein the one of the different instruction sequences is located at the program address, and wherein the random number is associated with one of the different instruction sequences via the program address.

13. (previously presented) The microcontroller of claim 1, wherein the instructions further comprise a jump instruction subject to a condition which if true directs the central processing unit to a program address, wherein the one of the different instruction sequences is located at the program address, and wherein the random number is associated with the one of the different instruction sequences via the condition.

14. (previously presented) The microcontroller of claim 1, wherein the random number is generated by a random number generator assigned to the microcontroller.